Article

A New Principle Toward Robust Matching in Human-like Stereovision

Ming Xie [1,*], Tingfeng Lai [2] and Yuhui Fang [2]

[1] School of Mechanical and Aerospace Engineering, Nanyang Technological University; mmxie@ntu.edu.sg
[2] School of MAE, Nanyang Technological University; LAIT0012@e.ntu.edu.sg
* Correspondence: mmxie@ntu.edu.sg; Tel.: +65 67905754

Abstract: Visual signals are the upmost important source for robots, vehicles or machines to achieve human-like intelligence. Human beings heavily depend on binocular vision to understand the dynamically changing world. Similarly, intelligent robots or machines must also have the innate capabilities of perceiving knowledge from visual signals. Until today, one of the biggest challenges faced by intelligent robots or machines is the matching in stereovision. In this paper, we present the details of a new principle toward achieving a robust matching solution which leverages on the use and integration of top-down image sampling strategy, hybrid feature extraction, and RCE neural network for incremental learning (i.e., cognition) as well as robust match-maker (i.e., recognition). A preliminary version of the proposed solution has been implemented and tested with data from Maritime RobotX Challenge (www.robotx.org). The contribution of this paper is to attract more research interest and effort toward this new direction which may eventually lead to the development of robust solutions expected by future stereovision systems in intelligent robots, vehicles and machines.

Keywords: Visual Signals; Stereovision; Image Sampling; Feature Extraction; Incremental Learning; Match-Maker; Cognition; Recognition; Possibility Function.

1. Introduction

We are living inside an ocean of signals. Among all the signals, visual signals should be the ones with the upmost importance. This is because without the visual signals, human beings will not be able to undertake many activities in the physical world. Similarly, visual signals are extremely important to today's autonomous robots, vehicles and machines [1]. Hence, research works on enabling robots, vehicles and machines to gain human-like intelligence from the use of visual signals should never be undermined or shadowed by the development of alternative sensors such as Radar [2] and LiDAR [3].

In this paper, we present a new principle which addresses the most difficult problem in stereovision, which is to achieve stereo matching as robust as possible [4]. The motivation behind our research works comes from projects dedicated to the development of intelligent humanoid robots [5] as well as autonomous surface vehicles [6], as shown in Figure 1.

For both platforms in Figure 1, their intelligence and autonomy greatly depend on the outer loop which consists of perception, planning and control. Among all possible modalities of doing perception, visual perception is a very important one. Especially, the goal toward achieving human-like visual perception must start with the use of binocular vision or stereovision [7]. Hence, research on human-like stereovision should be a non-negligible topic in both artificial intelligence and robotics. Therefore, the purpose of this paper is to present a new principle which advances the current state of the art in developing human-like stereovision for autonomous robots, vehicles, and machines [8].

The remaining part of the paper is organized as follows: Section 2 outlines the biggest challenge faced by stereovision. Section 3 briefly discusses similar works dedicated to stereo matching. Section 4 describes the proposed new principle which addresses stereo matching problem. Sections 5 to 10 present the details of the key steps inside the proposed principle. Section 11 shows some preliminary results. The conclusions are given in Section 12.

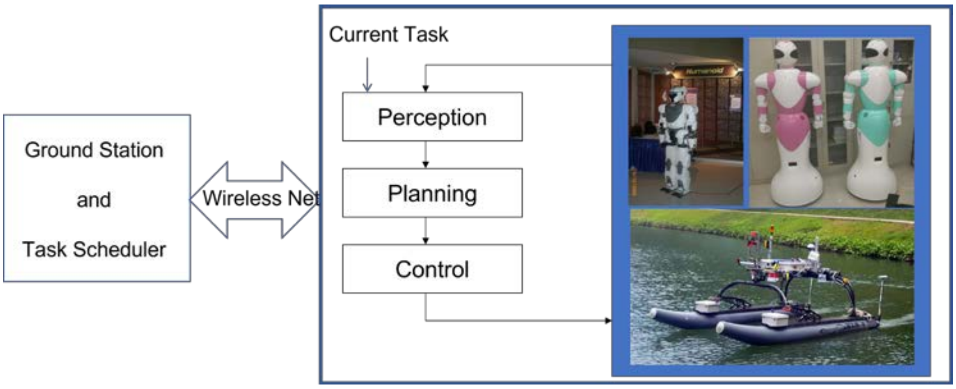


Figure 1. Research Framework Underlying the Development of Intelligent Humanoid Robots and Autonomous Surface Vehicles.

2. Problem Statement

Stereovision enables human beings to classify entities, to identify entities, and to localize entities. Clearly, stereovision is a very powerful system or module which provides answers to the following questions:

1. What are the classes of perceived entities?
2. What are the identities of perceived entities?
3. Where are the locations of perceived entities inside related images?
4. Where are the locations of perceived entities inside scenes?

A complete solution, which could fully answer the above questions, actually depends on the availability of the working principles underlying image sampling (NOTE: this is a largely overlooked sub-topic), entity cognition (NOTE: vaguely named as deep learning in literature [9]), entity recognition (NOTE: vaguely named as object detection in literature [10]), and entity matching [11] as shown in Figure 2, etc.

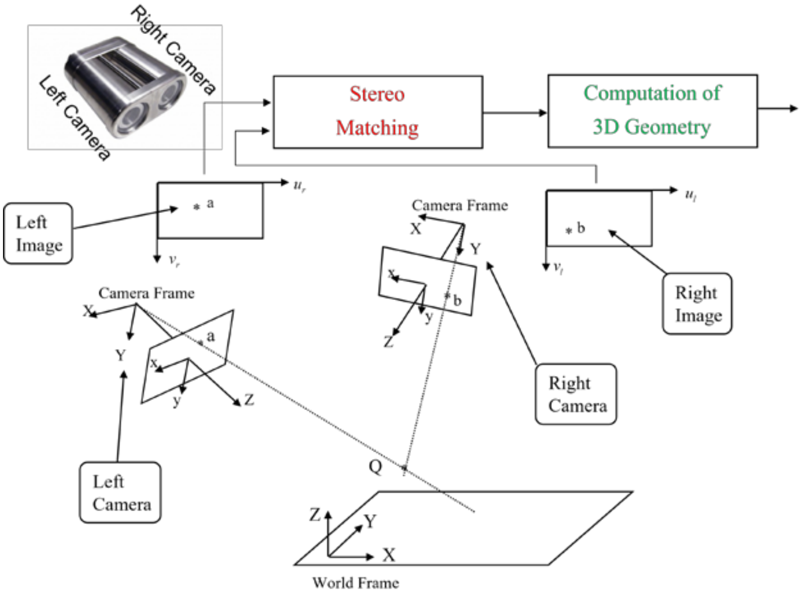


Figure 2. Illustration of Stereo Matching.

As illustrated in Figure 2, when an entity is placed at location Q in a scene, its stereo images will appear at location a in left image and location b in right image, respectively. Then, the stereo matching problem could be stated as follows:

Given the location of an entity seen by the left camera, how to determine the location of the same entity seen by the right camera? [12]

The above problem implies the following two challenges:

1. How to determine the presence of an entity in the left camera's image plane?
2. How to find the match in the right camera's image plane if an entity has been detected in the left camera's image plane?

It is important to further pay attention to the root causes behind these two challenges. The major root causes include [13]:

1. Variations of entities in size
2. Variations of entities in orientation
3. Variations of entities' images due to lighting conditions
4. Variations of entities' images due to occlusions as shown in Figure 3(a)
5. Variations of entities' images due to image sampling process as shown in Figure 3(b)

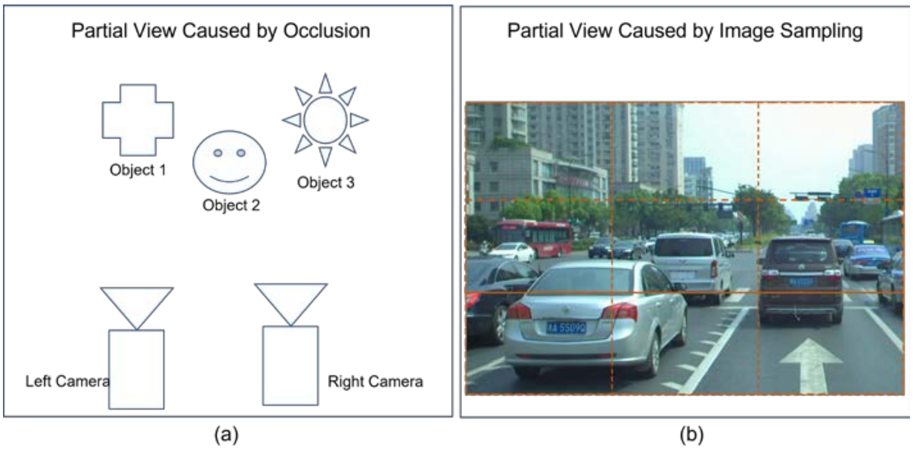

Partial View Caused by Occlusion
Object 1
Object 2
Object 3
Left Camera
Right Camera
(a)
Partial View Caused by Image Sampling
(b)

Figure 3. Illustration of (a) partial view caused by occlusion in which left camera sees partial view of object 3 while right camera sees partial view of object 1, and (b) partial view caused by image sampling in which the three nearest vehicles partially appear inside samples at row 2 and row 3.

In practice, we could cope with the issues raised by the variations of images in sizes and orientations if we could afford to have enough computational powers allocated to process images at multiple scales and multiple rotations. Also, we could cope with the variations of lighting conditions if we could make use of cameras with built-in functions of automatic illumination compensation and/or automatic contrast equalization. Hence, our remaining effort should focus on dealing with the issue raised by the occurrence of partial views faced by stereovision.

3. Similar Works on Stereo Matching

Stereo matching is an old problem in computer vision. In literature, there is a tremendous amount of works dedicated to solving the problem faced by stereo matching. For example, there are:

1. Methods which make the attempt of matching points within a pair of stereo images [14].
2. Methods which make the attempt of matching edges or contours within a pair of stereo images [15].

3. Methods which make the attempt of matching line segments within a pair of stereo images [16].
4. Methods which make the attempt of matching curves within a pair of stereo images [17].
5. Methods which make the attempt of matching regions within a pair of stereo images [18].
6. Methods which make the attempts of matching objects within a pair of stereo images [19].

The proposed principle in this paper falls into the category of making the attempt of matching entities within a pair of stereo images. Here, an entity may broadly refer to an object, a person, an animal, a building, or a machine, etc. In the literature, the existing solution in this category focuses on the use of deep convolution to do feature extraction which is then followed by the use of artificial neural network to do tuning and prediction. Such methods actually depend on the process of bottom-up optimization (e.g., back propagation algorithm) and the use of features in time-domain. In contrast, our proposed principle advocates the use of top-down design process in which we promote the use of hybrid features (i.e., features from both time-domain and frequency domain) as well as the use of the improved version of RCE neural work [20]. RCE neural network [21-23], which was discovered in 1970s by a research team led by a laureate [23] of Nobel prize in 1969, is fundamentally different from artificial neural network. So far, to the best of our knowledge, there is no other better way of designing human-like cognition and recognition than the use of RCE neural work or its improved versions [20-23].

It is worth acknowledging that despite the huge amount of research works dedicated to stereovision, the achieved results are far behind the performance of human beings' stereovision. Obviously, we should not stop the continuous investigation which aims at looking for better principles of, or solutions to, human-like stereovision.

4. The Outline of Proposed Principle

Human vision is attention-driven in a top-down manner. The attention could be triggered by the occurrences of reference entities such as appearances of persons, appearances of animals, appearances of objects, appearances of machines, appearances of geometries (e.g., lines, curves, surfaces, volumes, etc), appearances of photometry (e.g., chrominance and luminance, etc), appearances of textures, etc. Such reference entities could be learnt by a cognition process incrementally in real-time. However, the occurrences of familiar reference entities should be the responses of an internal recognition process.

Inspired by the innate processes of human vision, we propose a new principle which imitates the attention-driven behavior of human vision. The main idea of the proposed new principle is outlined in Figure 4.

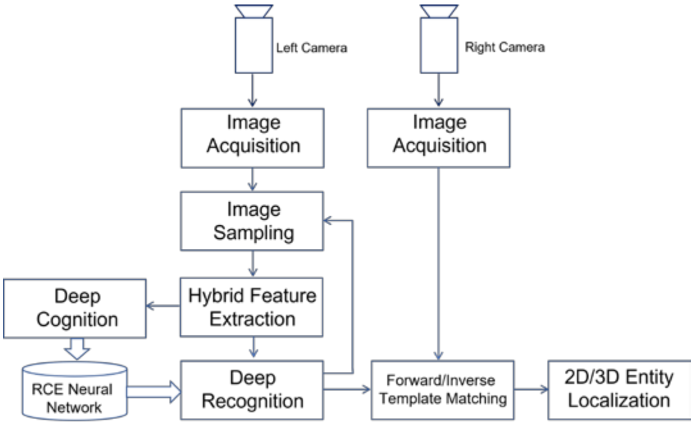


Figure 4. Outline of Proposed New Principle Toward Achieving Robust Matching in Human-like Stereovision.

Without loss of generality, we assume that the attention is to be recognized from the video streams of the left camera. The key steps involved in the proposed new principle include:

1. Image acquisition by both cameras.
2. Image sampling on video stream from left camera.
3. Hybrid feature extraction for each image sample.
4. Cognition of image samples if they correspond to the training data of reference entities inside training images.
5. Recognition of image samples if they correspond to the possible occurrences of reference entities inside real-time images.
6. Forward/Inverse processes of template matching, which work together so as to find the occurrence of matched candidate in the right image, if a recognized entity is present in the left image.

In the subsequent sections, we will describe the details of key steps 2 to 6.

5. Top-Down Strategy of Doing Image Sampling

An image may contain many entities of interest. One of the biggest challenges faced by image understanding or image segmentation/grouping is to divide an image into a matrix of image samples, each of which just contains the occurrence or appearance of a single entity. In theory and in practice, there is no solution which could generally guarantee such results expected by the subsequent visual processes in stereovision.

In addition, the problem of finding better ways to do image sampling did not receive enough attention in the research community. One major reason is because many people believe that it is good enough to use a sub-window to scan an input image so as to obtain all the possible image samples. However, this way of doing image sampling has serious drawbacks such as:

1. It is difficult to determine, or to justify, the size of sub-window which is used to scan an input image. If the size of sub-window is allowed to be dynamically changed, then the next question is how to do such dynamic adjustment of sizes.
2. The number of obtained image samples is independent of the content inside an input image. For example, an input image may contain a single entity. In this case, the scanning method will still produce many image samples which will be the input to subsequent visual processes of classification, identification, and grouping, etc. Obviously, irrelevant image samples may potentially cause troubles to these visual processes of recognition.

In this paper, we advocate a top-down strategy which iteratively divides an input image into a list of sets which contain linearly growing numbers of image samples of different sizes. If we denote S_k a set which contains k image samples, one way to obtain S_k is to uniformly divide an input image into a matrix of $d_v \times d_h$ samples, in which $d_v \times d_h = k$. For example, if we iteratively divide an input image into:

1. S_k with one sample, then $k = 1$ and $d_v \times d_h \in [1 \times 1]$.
2. S_k with two samples, then $k = 2$ and $d_v \times d_h \in [1 \times 2, 2 \times 1]$.
3. S_k with three samples, then $k = 3$ and $d_v \times d_h \in [1 \times 3, 3 \times 1]$.
4. S_k with four samples, then $k = 4$ and $d_v \times d_h \in [1 \times 4, 4 \times 1, 2 \times 2]$.
5. and so on.

This top-down strategy of doing image sampling is suitable for both parallel implementation and sequential implementation.

Before sending the image samples to the next visual process of extracting features, it is necessary to normalize the size of image samples so as to make them to be comparable in size. In practice, it is trivial to scale up or down the size of an image sample to any chosen standard value. By now, we could represent image sample set S_k as follows:

$$S_k = \{I_{j,r}(u,v), I_{j,g}(u,v), I_{j,b}(u,v), u \in [0, U-1], v \in [0, V-1], j \in [1, k]\} \quad (1)$$

where (r, g, b) are the three primary color components at index coordinates (u, v) inside set S_k's j^{th} image sample $I_j(u, v)$ which has the size of $V \times U$. Hence, by default, each image acquisition module in stereovision outputs color images, each of which is represented by a set of three matrices such as $I_{j,r}(u, v), I_{j,g}(u, v), I_{j,b}(u, v)$ in Equation 1.

6. Feature Extraction from Sample Image in Time-Domain

Mathematically speaking, the periodicity in space is equivalent to the periodicity in time. Hence, without loss of generality, we consider the spatial axes of an image or image sample as time axes. In this way, we could focus our discussions on how to extract features in time domain as well as in frequency domain.

Feature extraction in time domain has been extensively investigated by the research community of image processing and computer vision. In general, the basic operations include the computations of n-order derivatives where n could be equal to 0, 1, 2, 3, and any other larger value of integer. Here, the zero-order derivatives could refer to the results obtained by the operation of image smoothing for noise reduction (e.g., to use Gaussian filters).

In the literature, there are also many advanced studies which explore the use of Laplacian filters, Gabor filters, Wavelet filters, Moravec corner filter, Harris-Stephens corner filter, and Shi-Tomasi corner filter, etc. Hence, feature extraction in time domain is a very rich topic.

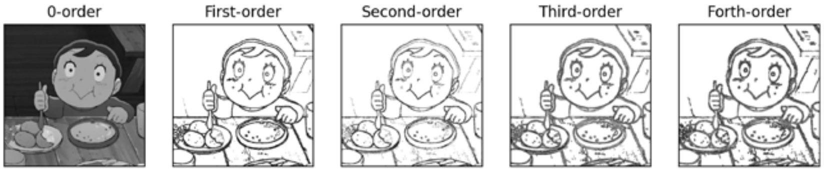


Figure 5. Examples of results from the computations of zero-order derivatives, first-order derivatives, second-order derivatives, third-order derivatives, and forth-order derivatives.

From the results shown in Figure 5, it is clear to us that higher order derivatives do not significantly provide extra information. The data of zero-order derivatives and first-order derivatives should be good enough for us to extract meaningful features in time domain.

In practice, the zero-order derivatives could be obtained by convoluting set S_k's j^{th} image sample $I_j(u, v)$ with a discrete Gaussian filter such as:

$$G(u, v) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, v \in [0,2], u \in [0,2] \tag{2}$$

If we represent the results (i.e., a matrix of zero-order derivatives of all the color components) of zero-order derivatives as follows:

$$S_{k,0} = \{I_{j,r_0}(u, v), I_{j,g_0}(u, v), I_{j,b_0}(u, v), u \in [0, U-1], v \in [0, V-1], j \in [1, k]\} \tag{3}$$

Then, the first-order derivatives could be obtained by convoluting each matrix in $S_{k,0}$ with the following two Sobel filters:

$$C_h(u, v) = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, v \in [0,2], u \in [0,2] \tag{4}$$

and

$$C_v(u,v)=\begin{bmatrix}+1 & 0 & -1\\ +2 & 0 & -2\\ +1 & 0 & -1\end{bmatrix}, v\in[0,2], u\in[0,2] \tag{5}$$

Clearly, the convolution with filter in Equation 4 will result in the horizontal components of the first-order derivatives while the convolution with filter in Equation 5 will result in the vertical components of the first-order derivatives. The L^2 norms computed from these two components will yield the results of the first-order derivatives of image samples $S_{k,0}$, which could be represented by:

$$S_{k,1}=\{I_{j,r_1}(u,v), I_{j,g_1}(u,v), I_{j,b_1}(u,v), u\in[0,U-1], v\in[0,V-1], j\in[1,\mathrm{k}]\} \tag{6}$$

Therefore, for image sample j in set S_k, it actually has six image matrices which are: $[I_{j,r_0}(u,v), I_{j,g_0}(u,v), I_{j,b_0}(u,v)]$ in Equation 3 and $[I_{j,r_1}(u,v), I_{j,g_1}(u,v), I_{j,b_1}(u,v)]$ in equation 6. Then, the next question is how to determine a feature vector F_j which meaningfully represents image sample j in set S_k.

A simple answer to the above question could be to convert the six image matrices of a sample into their vector representations (i.e., a 2D matrix is re-arranged as a 1D vector). Then, by putting these six image vectors together, we will obtain feature vector F_j. The advantage of this method is its simplicity. However, the noticeable drawback is the large dimension of feature vector F_j. Then, we may want to know whether there is a better way of determining feature vector F_j from image matrices, or not.

So far, there is no theoretical answer to this question. Maybe, a practical way is to design workable solutions which could be suitable for applications in hands. In this way, a library of workable solutions may empower autonomous robots, vehicles, or machines to adapt their behaviors to real-time situations or applications. Clearly, this topic still offers opportunities for further or continuous research works.

Here, we propose a simple and practical way of determining feature vector F_j from image matrices. The idea is to compute statistics from a set of image matrices. Interestingly, the two obvious types of statistics are the mean values and standard deviations.

For example, if $\{I(u,v), u\in[0,U-1], v\in[0,V-1]\}$ is an image matrix of single values such as red components, green components, blue components, or their individual first-order derivatives, each value in $\{I(u,v), u\in[0,U-1], v\in[0,V-1]\}$ could be considered as a kind of measurement of approximate electromagnetic energy. Therefore, we could compute the following four meaningful statistics from $\{I(u,v), u\in[0,U-1], v\in[0,V-1]\}$, which are:

1. The mean value of approximate electromagnetic energy:

$$I_a=\frac{1}{UV}\sum_{v=0}^{V-1}\sum_{u=0}^{U-1}I(u,v) \tag{7}$$

2. The square-root of the variance of approximate electromagnetic energy:

$$\sigma_I=\sqrt{\frac{\sum_{v=0}^{V-1}\sum_{u=0}^{U-1}(I(u,v)-I_a)^2}{UV}} \tag{8}$$

3. The horizontal distribution of approximate electromagnetic energy:

$$\sigma_u=\sqrt{\frac{\sum_{v=0}^{V-1}\sum_{u=0}^{N-1}I(u,v)\times(u-u_c)^2}{\sum_{v=0}^{V-1}\sum_{u=0}^{U-1}I(u,v)}} \tag{9}$$

with:

$$u_c=\frac{\sum_{v=0}^{V-1}\sum_{u=0}^{N-1}\{I(u,v)\times u\}}{\sum_{v=0}^{V-1}\sum_{u=0}^{U-1}I(u,v)} \tag{9a}$$

and

$$v_c = \frac{\sum_{v=0}^{V-1}\sum_{u=0}^{N-1}\{I(u,v) \times v\}}{\sum_{v=0}^{V-1}\sum_{u=0}^{U-1} I(u,v)} \tag{9b}$$

4. The vertical distribution of approximate electromagnetic energy:

$$\sigma_v = \sqrt{\frac{\sum_{v=0}^{V-1}\sum_{u=0}^{N-1}\{I(u,v) \times (v - v_c)^2\}}{\sum_{v=0}^{V-1}\sum_{u=0}^{U-1} I(u,v)}} \tag{10}$$

As a result, any image matrix such as $\{I(u,v), u \in [0, U-1], v \in [0, V-1]\}$ could be represented by feature vector F as follows:

$$F = [I_a, \sigma_I, \sigma_u, \sigma_v] \tag{11}$$

In time domain, if image sample j in set S_k has six image matrices, its feature vector F_j will contain 24 feature values.

7. Feature Extraction from Sample Image in Frequency-Domain

In mathematics, a very important discovery was Fourier Transform which tells us that any signal is the (finite or infinite) sum of sine functions. In engineering, one of the greatest inventors was Nikolas Tesla who told us that the secret of the universe could be understood by simply thinking in terms of energy, vibration, and frequency. Such statement explicitly advises us to look for feature space and feature vector in frequency domain if we would like to understand the secret of machine intelligence.

Given image matrix $\{I(u,v), u \in [0, U-1], v \in [0, V-1]\}$, it could be represented by, or decomposed into, its Fourier series in terms of complex exponentials $e^{\pm ix}$ (in which $i = \sqrt{-1}$) which could be computed as follows:

$$I(u,v) = \frac{1}{UV}\sum_{\omega_v=0}^{V-1}\sum_{\omega_u=0}^{U-1} \hat{I}(\omega_u,\omega_v) e^{i\left(\frac{2\pi\omega_u u}{U}\right)} e^{i\left(\frac{2\pi\omega_v v}{V}\right)} \tag{12}$$

with $0 \le u \le U-1$, $0 \le v \le V-1$ and:

$$\hat{I}(\omega_u,\omega_v) = \sum_{v=0}^{V-1}\sum_{u=0}^{U-1} I(u,v) e^{-i\left(\frac{2\pi\omega_u u}{U}\right)} e^{-i\left(\frac{2\pi\omega_v v}{V}\right)} \tag{13}$$

in which $0 \le \omega_u \le U-1$ and $0 \le \omega_v \le V-1$. With continuous signals or data, Equation 12 will become inverse Fourier Transform while Equation 3 will become forward Fourier transform.

It is interesting to take note that each value $\hat{I}(\omega_u,\omega_v)$ in Equation (13) is a complex number or more precisely a vector. Mathematically speaking, a vector indicates a position in a space. Hence, Fourier coefficient vectors (or complex numbers), which are stored inside complex matrix $\{\hat{I}(\omega_u,\omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$, nicely define a feature space. Such a feature space could be called as Fourier feature space.

In mathematics, complex matrix $\{\hat{I}(\omega_u,\omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$ could be split into two ordinary matrices $\{A(\omega_u,\omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$ and $\{B(\omega_u,\omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$, where $A(\omega_u,\omega_v)$ is the real part of complex number (or vector) $\hat{I}(\omega_u,\omega_v)$ and $B(\omega_u,\omega_v)$ is the imaginary part of complex number (or vector) $\hat{I}(\omega_u,\omega_v)$. Both matrices A and B are Fourier coefficient matrices.

Therefore, in frequency domain, a straightforward way of determining feature vector F which characterizes image matrix $\{I(u,v), u \in [0, U-1], v \in [0, V-1]\}$ taken from image sample j in set S_k is to re-arrange the corresponding Fourier coefficient matrix $\{A(\omega_u,\omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$ or $\{B(\omega_u,\omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$ into a vector.

doi:10.20944/preprints202306.1313.v1

Alternatively, we could use equations, which are the same to Equations (7) to (10), to compute the mean values $(\omega_{u,c}, \omega_{v,c})$ of frequencies and their standard deviations $(\sigma_{\omega,u}, \sigma_{\omega,v})$ from Fourier coefficient matrix $\{A(\omega_u, \omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$ or $\{B(\omega_u, \omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$. In this way, frequency domain's feature vector corresponding to each Fourier coefficient matrix $\{A(u, v), u \in [0, U-1], v \in [0, V-1]\}$ or $\{B(\omega_u, \omega_v), \omega_u \in [0, U-1], \omega_v \in [0, V-1]\}$ could be as follows:

$$F_j = [\omega_{u,c}, \omega_{v,c}, \sigma_{\omega,u}, \sigma_{\omega,v}] \tag{14}$$

In time domain, each image sample j in set S_k has three color component images. Each color component image could yield two Fourier coefficient matrices. In total, there will be six Fourier coefficient matrices for any given image sample j in set S_k. As a result, in frequency domain, feature vector F_j of image sample j in set S_k will also contain 24 feature values.

8. Cognition Process Using RCE Neural Network

Today, many researchers still believe that our mind arises from our brain. This opinion makes a lot of people or young researchers believe that the blueprint of mind is part of the blueprint of brain. For those who are familiar with microprocessors and operating systems, it is clear to us that the blueprints of operating systems are not part of the blueprints of microprocessors.

Here, we advocate the truth which states that mind is mind while brain is brain. Most importantly, the basic functions of brain are to support memorizations and computations which are intended by mind. With this truth in mind, the future research in artificial intelligence or machine intelligence should be focused on the physical principles behind the design of human-like minds which could transform signals into the cognitive states of knowing the conceptual meanings behind the signals.

In the previous sections, we have discussed the details of feature extraction. The results are lists of feature vectors in time domain, frequency domain, or both. Then, the next question will be how to learn the conceptual meaning behind a set of feature vectors corresponding to the same class of sample images or the same identity of sample images. The good news is that RCE neural network discovered in 1970s provides us a better version of answers, so far.

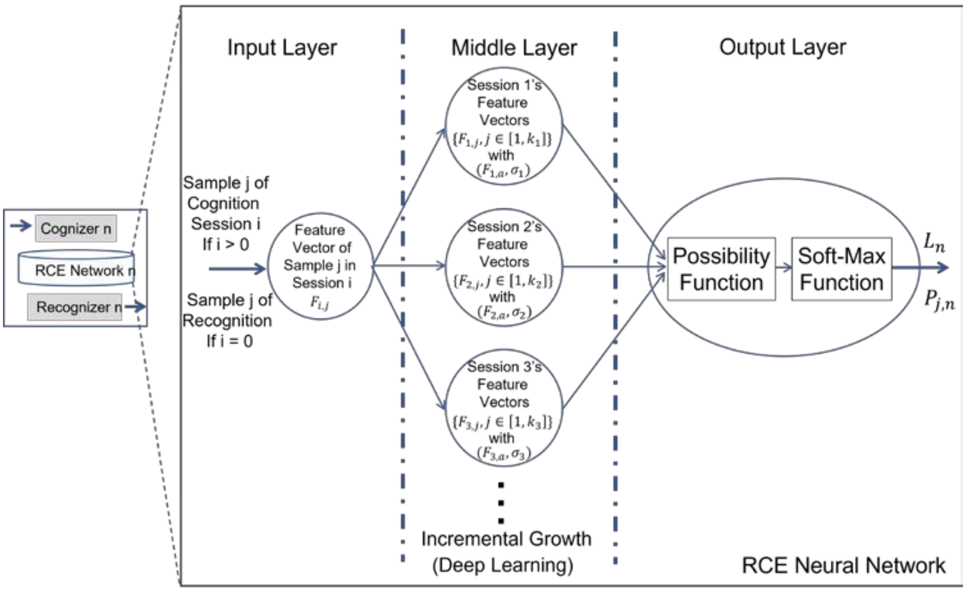

Cognizer n
RCE Network n
Recognizer n
Input Layer
Middle Layer
Output Layer
Sample j of Cognition Session i If i > 0
Sample j of Recognition If i = 0
Feature Vector of Sample j in Session i $F_{i,j}$
Session 1's Feature Vectors $\{F_{1,j}, j \in [1, k_1]\}$ with $(F_{1,a}, \sigma_1)$
Session 2's Feature Vectors $\{F_{2,j}, j \in [1, k_2]\}$ with $(F_{2,a}, \sigma_2)$
Session 3's Feature Vectors $\{F_{3,j}, j \in [1, k_3]\}$ with $(F_{3,a}, \sigma_3)$
Incremental Growth (Deep Learning)
Possibility Function
Soft-Max Function
L_n
$P_{j,n}$
RCE Neural Network

Figure 6. Structure of RCE Neural Network.

As shown in Figure 6, both cognition and recognition could be implemented with the use of RCE neural network which consists of three layers. There is a single vector at the input layer. Also, there is a single vector at the output layer. However, inside the middle layer, there is a dynamically growing number of nodes, each of which memorizes the feature vectors from a set of sample images provided by a training session of cognition.

Clearly, RCE neural network is fundamentally different from the so-called artificial neural network which is simply a graphical representation of a system of equations with coefficients to be tuned in some simple or deep manners (e.g., back-propagation method).

Refer to Figure 6. With training session i's feature vectors, we could easily compute the mean vector and the standard deviation of the distances from the training session's feature vectors to their mean vector.

For example, if training session i has k_i sample images which form the following set:

$$S_{k_i} = \{I_{j,r}(u,v), I_{j,g}(u,v), I_{j,b}(u,v), u \in [0, U-1], v \in [0, V-1], j \in [1, k_i]\} \tag{15}$$

then training session i's set of feature vectors computed by feature extraction module could be denoted by $\{F_{i,j}, j \in [1, k_i]\}$ where j is the index of image sample j in set S_{k_i}. Subsequently, the mean vector of $\{F_{i,j}, j \in [1, k_i]\}$ could be calculated by:

$$F_{i,a} = \frac{1}{k_i}\sum_{j=1}^{k_i} F_{i,j} \tag{16}$$

and the standard deviation of the distances from $\{F_{i,j}, j \in [1, k_i]\}$ to the mean vector could be computed by:

$$\sigma_i = \sqrt{\frac{1}{k_i}\sum_{j=1}^{k_i}(F_{i,j} - F_{i,a})^T(F_{i,j} - F_{i,a})} \tag{17}$$

By now, we could explain he physical meaning of node i (i.e., outcome of training session i) in RCE neural network, which is simply the representation of hyper-sphere [21] with its center at $F_{i,a}$ and its radius to be equal to $3\sigma_i$. Since i could dynamically grow, RCE neural network naturally supports the process of incrementally learning as well as the process of deep learning which is widely discussed about in the literature.

As we mentioned above, the deep tuning of parameters inside a complex artificial neural network, which is a graphical representation of a system of equations, has nothing to do with deep learning, and the true nature of deep learning is outlined in Figure 6.

In summary, a training session for cognizing entity n consists of supplying a set of entity n's image samples in Equation (15) and entity n's conceptual meaning L_n which is a label or a word in a natural language such as English.

9. Recognition Process Using Possibility Function

Refer to Figure 6 again. With a trained RCE neural network by a cognition process for each entity of interest (e.g., entity n), the output layer is primarily for the purpose of executing recognition process when the feature vector computed from any arbitrary image sample is given to the input layer.

In the literature, many researchers believe that recognition is a process of determining the chances of occurrences. As a result, probability functions are widely used inside a recognition module.

Here, we advocate the truth which states that recognition is a process of evaluating the beliefs about the identities and categories of any arbitrary image sample at input. This truth is in line with the fact that our mind consists of many sub-systems of beliefs. Hence, the function for estimating the degrees of beliefs should be a probability function such as:

$$p_{j,n}(i) = e^{-\frac{1}{2\sigma_i^2}(F_{0,j}-F_{i,a})^T(F_{0,j}-F_{i,a})} \tag{18}$$

where $(F_{i,a}, \sigma_i)$ is the parameter vector of the hyper-sphere obtained from training session i while $F_{0,j}$ is the feature vector computed from image sample j during recognition process, and $p_{j,n}(i)$ is the possibility for image sample j to belong to learnt entity n according to training session i's parameter vector.

Since RCE neural network intrinsically supports incremental learning as well as deep learning, the single node in the output layer must include a Soft-Max function such as:

$$P_{j,n} = \max_i\{p_{min}, p_{j,n}(i)\} \tag{19}$$

where p_{min} is the minimum value of acceptable possibility (e.g., 0.5). In practice, if $P_{j,n} = p_{min}$, the interpretation could be stated as follows: input $F_{0,j}$ does not support the belief that image sample j belongs to learnt entity n. Otherwise, if $P_{j,n} > p_{min}$, it means that the output of recognition will be $(L_n, P_{j,n})$ in which L_n is the conceptual meanings of image sample j.

10. Forward/Inverse Processes of Template Matching

In the previous sections, we have discussed the key details about the modules of image sampling, hybrid feature extraction, cognition, and recognition. In a human-like stereovision system, these modules will produce the output of recognized entities inside left camera's image plane, as illustrated by Figure 4. Then, the next question will be how to determine the match in right camera if a recognized entity in left camera is given. This question describes the famous problem of stereo matching faced by today's stereovision systems.

In the literature, stereo matching is a widely investigated problem. So far, there is no solution which could achieve the performance close to, or as good as, the one of human being's stereovision system. Hence, better solutions for improved performance are still expected from future research works in this area.

In this paper, we present a new strategy which could cope with the problem of stereo matching in a better way. This new strategy consists of the interplay between forward template matching and inverse template matching.

In stereovision, the only geometrical constraint is the so-called epipolar line which indicates the possible locations of a match (e.g., at location b in Figure 7) in right image plane if a location in left image plane is given (e.g., location a in Figure 7).

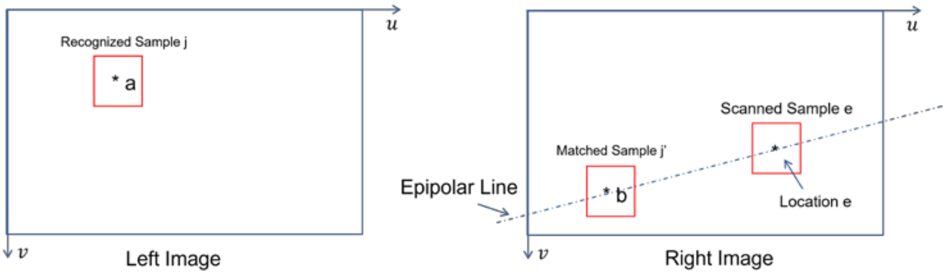


Figure 7. Illustration of Forward Template Matching in Stereovision.

As shown in Figure 7, if recognized sample j at location a is given in left image plane, the forward process of template matching will consist of the following steps:

1. Determine the equation of epipolar line from both stereovision's calibration parameters (NOTE: such knowledge could be found in any textbook of computer vision) and location a's coordinates.
2. Scan the epipolar line location by location.
3. Take image sample e at currently scanned location e.

4. Compute the feature vector of image sample e.
5. Compute the cosine distance between image sample j's feature vector and image sample e's feature vector.
6. Repeat the scanning until it is completed.
7. Choose the image sample to be the candidate of matched sample j' if it minimizes the cosine distance.
8. Use the cosine distance between recognized sample j and the chosen candidate of matched sample j' to compute the possibility value of match (i.e., to use Equation 18).
9. Accept matched sample j' if the possibility value of match is greater than a chosen threshold value (e.g., 0.5).

In the above process, if F_j is the feature vector of image sample j while F_e is the feature vector of image sample e, the cosine distance $d_{j,e}$ between those two vectors is simply calculated according to their inner product, which is:

$$d_{j,e} = \frac{F_j^T \times F_e}{\|F_j\| \times \|F_e\|} \tag{20}$$

and the corresponding possibility value is calculated as follows:

$$P_{j,e} = e^{-\frac{1}{2\sigma_0^2} d_{j,e}^2} \tag{21}$$

where σ_0 is a default value of standard deviation which could be self-determined by robots, vehicles, or machines during a training session of cognition process.

According to the illustration shown in Figure 3, the forward process of template matching will work only if there is no partial view due to either occlusion or image sampling. If matched sample j' in right image contains partial view of recognized sample j in left image, the inverse process of template matching will perform better than its counterpart of forward process.

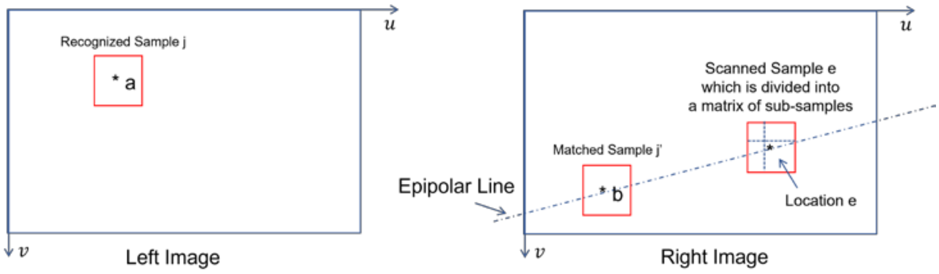


Figure 8. Illustration of Inverse Template Matching in Stereovision.

As shown in Figure 8, if recognized sample j at location a is given in left image plane, the inverse process of template matching consists of the following steps:

1. Determine the equation of epipolar line from both the stereovision's calibration parameters and the location a's coordinates.
2. Scan the epipolar line location by location.
3. Take image sample e at currently scanned location e.
4. Divide image sample e into a matrix of sub-samples $\{e_i, i = 1,2,3, \ldots\}$.
5. Use each sub-sample in $\{e_i, i = 1,2,3, \ldots\}$ as template and do forward template matching with recognized sample j.
6. Compute the mean value of all the possibility values which measure the match between all the sub-samples in $\{e_i, i = 1,2,3, \ldots\}$ and recognized sample j. This mean value represents the possibility value for image sample e in right image to match with recognized sample j in left image.
7. Repeat the scanning until it is completed.

doi:10.20944/preprints202306.1313.v1

8. Choose the image sample to be the candidate of matched sample j' if it minimizes the possibility values of match (i.e., calculated by Equation 21).
9. Accept the match if the possibility value of match is greater than a chosen threshold value (e.g., 0.5).

In practice, we could run both forward process and inverse process of template matching in parallel. In this way, a better decision of match in right image could be made if recognized sample j in left image is given.

11. Implementation and Results

The proposed new principle has been implemented in Python. Preliminary tests have been with image data from public domain. Especially, we use image data which are posted to public domain by maritime RobotX challenge (www.robotx.org). Figure 9 shows two typical examples of scenes constructed by maritime RobotX challenge.

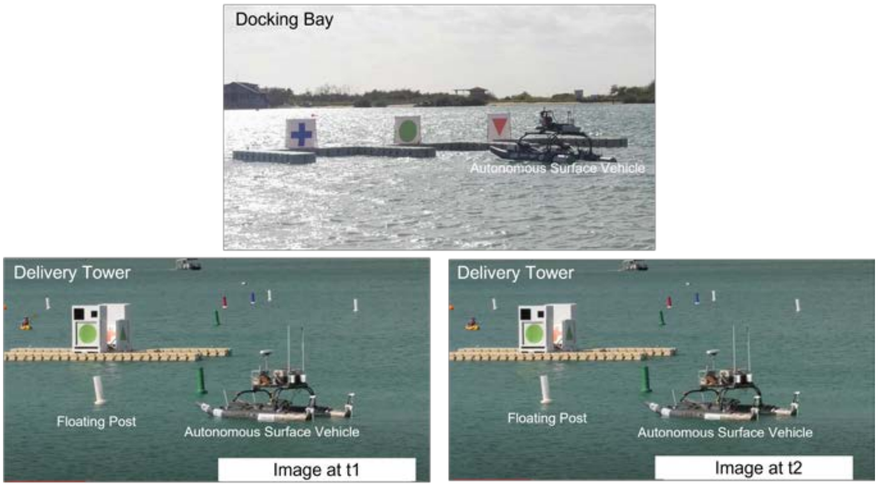


Figure 9. Typical scenes constructed by maritime RobotX challenge.

The tasks to be undertaken by an autonomous surface vehicle include stereovision-guided delivery of objects, stereovision-guided parking into the docking bay, etc. In the following sections, we share some of our experimental results.

11.1. Results of Top-down Sampling Strategy of Input Images

Our proposed top-down sampling strategy of input images (e.g., images from left camera) is to divide an input image from left camera into a list of sets S_k which contain increasing number of image samples (i.e., k = 1, 2, 3, …). Figure 10 shows an example of results (i.e., k = 28) from our proposed top-down sampling strategy of an input image. At this level of sampling, a red floating post clearly appears inside one of these 28 samples.

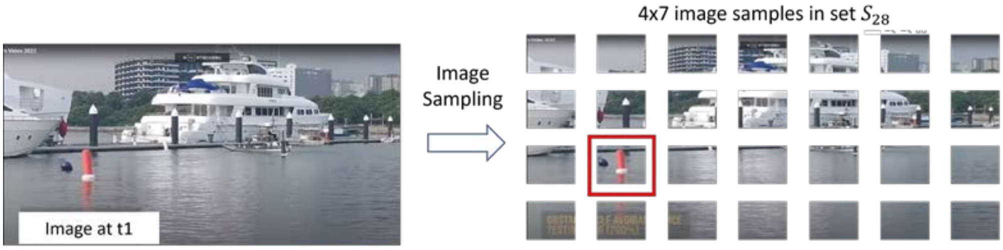


Figure 10. One Result of Top-down Sampling Strategy of Input Image.

doi:10.20944/preprints202306.1313.v1

11.2. Examples of Training Data for Cognition (i.e., Learning)

The proposed new principle involves the use of cognition and recognition modules. For cognition module, it is necessary to train it with training data of reference entities. Without loss of generality, we simply use a set of 10 samples to train the cognition module which is specifically dedicated to an entity of interest. It is amazing to see that the proposed solution could achieve successful results with 10 samples inside a dataset of training for each entity of interest.

Figure 11 shows the scenario of autonomous parking into a docking bay by an autonomous surface vehicle. In this task, the mental capabilities of the autonomous surface vehicle include a) cognition of triangle, cross and circle, and b) recognition of triangle, cross and circle. Hence, for the training of cognition module dedicated to each entity among triangle, cross and circle, we simply take ten samples as shown in Figure 11.

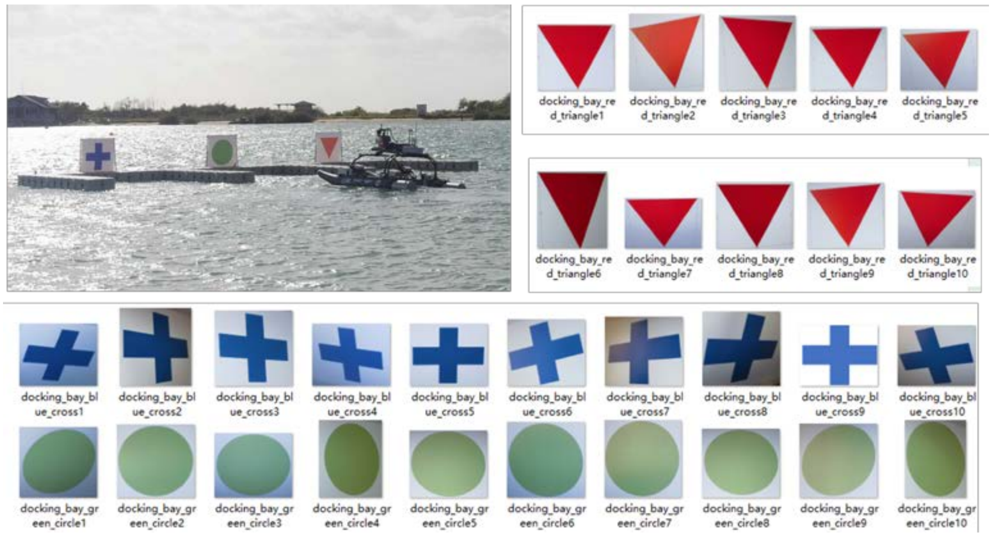
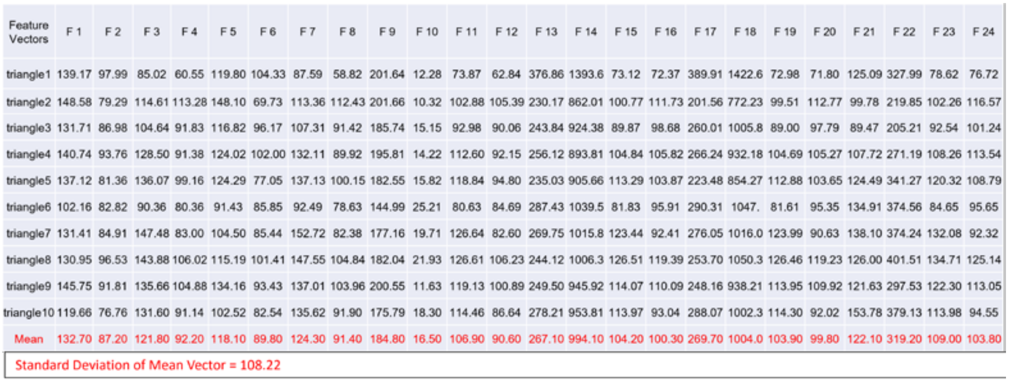

docking_bay_red_triangle1
docking_bay_red_triangle2
docking_bay_red_triangle3
docking_bay_red_triangle4
docking_bay_red_triangle5
docking_bay_red_triangle6
docking_bay_red_triangle7
docking_bay_red_triangle8
docking_bay_red_triangle9
docking_bay_red_triangle10
docking_bay_blue_cross1
docking_bay_blue_cross2
docking_bay_blue_cross3
docking_bay_blue_cross4
docking_bay_blue_cross5
docking_bay_blue_cross6
docking_bay_blue_cross7
docking_bay_blue_cross8
docking_bay_blue_cross9
docking_bay_blue_cross10
docking_bay_green_circle1
docking_bay_green_circle2
docking_bay_green_circle3
docking_bay_green_circle4
docking_bay_green_circle5
docking_bay_green_circle6
docking_bay_green_circle7
docking_bay_green_circle8
docking_bay_green_circle9
docking_bay_green_circle10

Figure 11. Ten Sample Images for Training Cognition Module Dedicated to Each Entity Among Triangle, Cross and Circle.

11.3. Results of Feature Extraction in Time Domain

For each sample image in Figure 10, we calculate its feature vector in time domain. Here, we share the results of feature vectors computed from the ten image samples of triangle in Figure 11. These results are shown in Figure 12, which also gives the result of the mean vector and its standard deviation.



Feature Vectors	F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F 10	F 11	F 12	F 13	F 14	F 15	F 16	F 17	F 18	F 19	F 20	F 21	F 22	F 23	F 24
triangle1	139.17	97.99	85.02	60.55	119.80	104.33	87.59	58.82	201.64	12.28	73.87	62.84	376.86	1393.6	73.12	72.37	389.91	1422.6	72.98	71.80	125.09	327.99	78.62	76.72
triangle2	148.58	79.29	114.61	113.28	148.10	69.73	113.36	112.43	201.66	10.32	102.88	105.39	230.17	862.01	100.77	111.73	201.56	772.23	99.51	112.77	99.78	219.85	102.26	116.57
triangle3	131.71	86.98	104.64	91.83	116.82	96.17	107.31	91.42	185.74	15.15	92.98	90.06	243.84	924.38	89.87	98.68	260.01	1005.8	89.00	97.79	89.47	205.21	92.54	101.24
triangle4	140.74	93.76	128.50	91.38	124.02	102.00	132.11	89.92	195.81	14.22	112.60	92.15	256.12	893.81	104.84	105.82	266.24	932.18	104.69	105.27	107.72	271.19	108.26	113.54
triangle5	137.12	81.36	136.07	99.16	124.29	77.05	137.13	100.15	182.55	15.82	118.84	94.80	235.03	905.66	113.29	103.87	223.48	854.27	112.88	103.65	124.49	341.27	120.32	108.79
triangle6	102.16	82.82	90.36	80.36	91.43	85.85	92.49	78.63	144.99	25.21	80.63	84.69	287.43	1039.5	81.83	95.91	290.31	1047.	81.61	95.35	134.91	374.56	84.65	95.65
triangle7	131.41	84.91	147.48	83.00	104.50	85.44	152.72	82.38	177.16	19.71	126.64	82.60	269.75	1015.8	123.44	92.41	276.05	1016.0	123.99	90.63	138.10	374.24	132.08	92.32
triangle8	130.95	96.53	143.88	106.02	115.19	101.41	147.55	104.84	182.04	21.93	126.61	106.23	244.12	1006.3	126.51	119.39	253.70	1050.3	126.46	119.23	126.00	401.51	134.71	125.14
triangle9	145.75	91.81	135.66	104.88	134.16	93.43	137.01	103.96	200.55	11.63	119.13	100.89	249.50	945.92	114.07	110.09	248.16	938.21	113.95	109.92	121.63	297.53	122.30	113.05
triangle10	119.66	76.76	131.60	91.14	102.52	82.54	135.62	91.90	175.79	18.30	114.46	86.64	278.21	953.81	113.97	93.04	288.07	1002.3	114.30	92.02	153.78	379.13	113.98	94.55
Mean	132.70	87.20	121.80	92.20	118.10	89.80	124.30	91.40	184.80	16.50	106.90	90.60	267.10	994.10	104.20	100.30	269.70	1004.0	103.90	99.80	122.10	319.20	109.00	103.80

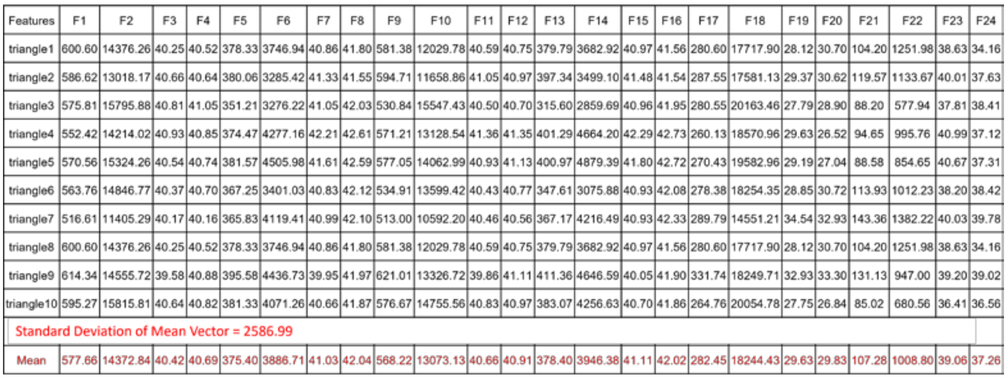
Standard Deviation of Mean Vector = 108.22

Figure 12. The Values of Ten Feature Vectors Computed from Ten Sample Images of Triangle in Time Domain.

doi:10.20944/preprints202306.1313.v1

11.4. Results of Feature Extraction in Frequency Domain

For each sample image in Figure 10, we calculate its feature vector in frequency domain. Similarly, we share the results of feature vectors computed from the ten image samples of triangle in Figure 11. These results are shown in Figure 13, which also gives the result of the mean vector and its standard deviation.



Features	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24
triangle1	600.60	14376.26	40.25	40.52	378.33	3746.94	40.86	41.80	581.38	12029.78	40.59	40.75	379.79	3682.92	40.97	41.56	280.60	17717.90	28.12	30.70	104.20	1251.98	38.63	34.16
triangle2	586.62	13018.17	40.66	40.64	380.06	3285.42	41.33	41.55	594.71	11658.86	41.05	40.97	397.34	3499.10	41.48	41.54	287.55	17581.13	29.37	30.62	119.57	1133.67	40.01	37.63
triangle3	575.81	15795.88	40.81	41.05	351.21	3276.22	41.05	42.03	530.84	15547.43	40.50	40.70	315.60	2859.69	40.96	41.95	280.55	20163.46	27.79	28.90	88.20	577.94	37.81	38.41
triangle4	552.42	14214.02	40.93	40.85	374.47	4277.16	42.21	42.61	571.21	13128.54	41.36	41.35	401.29	4664.20	42.29	42.73	260.13	18570.96	29.63	26.52	94.65	995.76	40.99	37.12
triangle5	570.56	15324.26	40.54	40.74	381.57	4505.98	41.61	42.59	577.05	14062.99	40.93	41.13	400.97	4879.39	41.80	42.72	270.43	19582.96	29.19	27.04	88.58	854.65	40.67	37.31
triangle6	563.76	14846.77	40.37	40.70	367.25	3401.03	40.83	42.12	534.91	13599.42	40.43	40.77	347.61	3075.88	40.93	42.08	278.38	18254.35	28.85	30.72	113.93	1012.23	38.20	38.42
triangle7	516.61	11405.29	40.17	40.16	365.83	4119.41	40.99	42.10	513.00	10592.20	40.46	40.56	367.17	4216.49	40.93	42.33	289.79	14551.21	34.54	32.93	143.36	1382.22	40.03	39.78
triangle8	600.60	14376.26	40.25	40.52	378.33	3746.94	40.86	41.80	581.38	12029.78	40.59	40.75	379.79	3682.92	40.97	41.56	280.60	17717.90	28.12	30.70	104.20	1251.98	38.63	34.16
triangle9	614.34	14555.72	39.58	40.88	395.58	4436.73	39.95	41.97	621.01	13326.72	39.86	41.11	411.36	4646.59	40.05	41.90	331.74	18249.71	32.93	33.30	131.13	947.00	39.20	39.02
triangle10	595.27	15815.81	40.64	40.82	381.33	4071.26	40.66	41.87	576.67	14755.56	40.83	40.97	383.07	4256.63	40.70	41.86	264.76	20054.78	27.75	26.84	85.02	680.56	36.41	36.56
Standard Deviation of Mean Vector = 2586.99																								
Mean	577.66	14372.84	40.42	40.69	375.40	3886.71	41.03	42.04	568.22	13073.13	40.66	40.91	378.40	3946.38	41.11	42.02	282.45	18244.43	29.63	29.83	107.28	1008.80	39.06	37.26

Figure 13. The Values of Ten Feature Vectors Computed from Ten Sample Images of Triangle in Frequency Domain.

11.5. Results of Cognition

The mean vector and its standard deviation, which are obtained from each training session for any entity of interest, will be stored inside a node at the middle layer of the RCE neural network which is allocated to an entity's cognition module. If there are N entities of interest, there will be a set of N RCE neural networks which support N pairs of cognizers and recognizers such as {(Cognizer n, Recognizer n), n = 1, 2, 3, …, N}. As illustrated in Figure 14, N could incrementally grow very deeply.

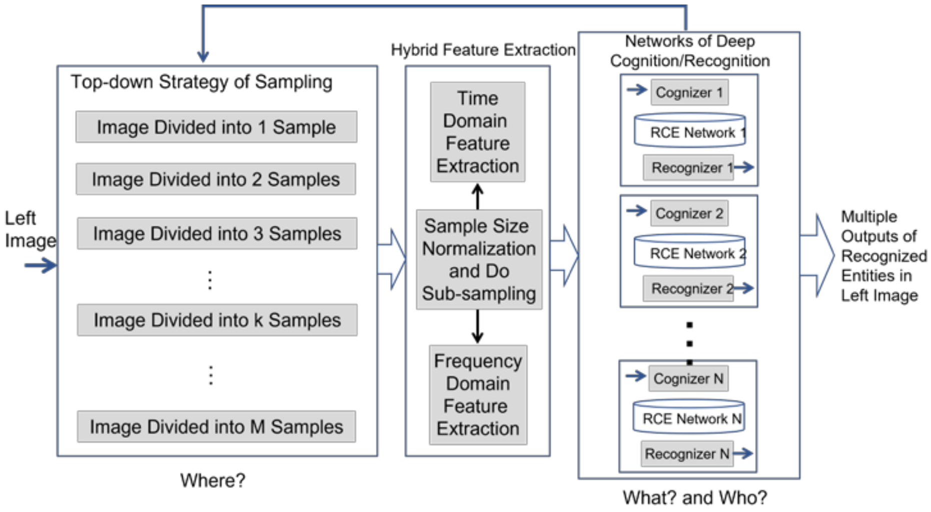

Top-down Strategy of Sampling
Image Divided into 1 Sample
Image Divided into 2 Samples
Image Divided into 3 Samples
⋮
Image Divided into k Samples
⋮
Image Divided into M Samples
Where?
Left Image
Hybrid Feature Extraction
Time Domain Feature Extraction
Sample Size Normalization and Do Sub-sampling
Frequency Domain Feature Extraction
Networks of Deep Cognition/Recognition
Cognizer 1
RCE Network 1
Recognizer 1
Cognizer 2
RCE Network 2
Recognizer 2
Cognizer N
RCE Network N
Recognizer N
What? and Who?
Multiple Outputs of Recognized Entities in Left Image

Figure 14. Results of Cognition in the Form of N Cognizers (i.e., 1, 2, 3, ..., N).

11.6. Results of Recognition

With N pairs of cognizers and recognizers such as {(Cognizer n, Recognizer n), n = 1, 2, 3, …, N} in place, an autonomous surface vehicle or robot is ready to recognize familiar or learnt entities inside images of left camera.

Figure 15 shows two examples of results of recognition in time domain. Each example contains seven image samples as input. Among these seven inputs, three of them are totally out of the class dedicated to the pair of cognizer and recognizer. We can see that recognition module performs quite successfully in recognizing the correct entries. Please take note that the feature vectors of image samples at input are all in time domain.

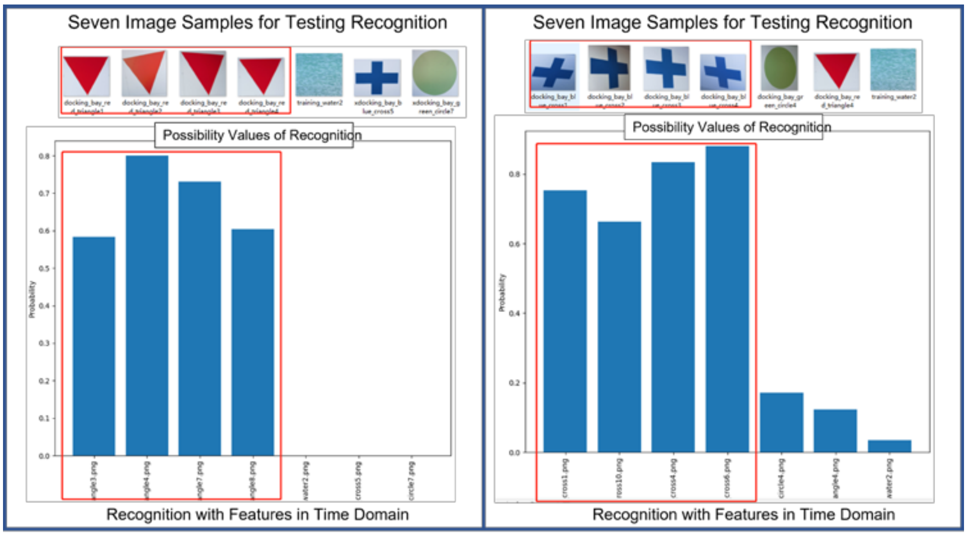


Figure 15. Two Examples of Results of Recognition Using Feature Vectors in Time Domain.

With the same image samples at input, Figure 16 shows the results of recognition in frequency domain. We can see that recognition module also performs quite successfully in recognizing the correct entries. Please take note that the feature vectors of image samples at input are all in frequency domain.

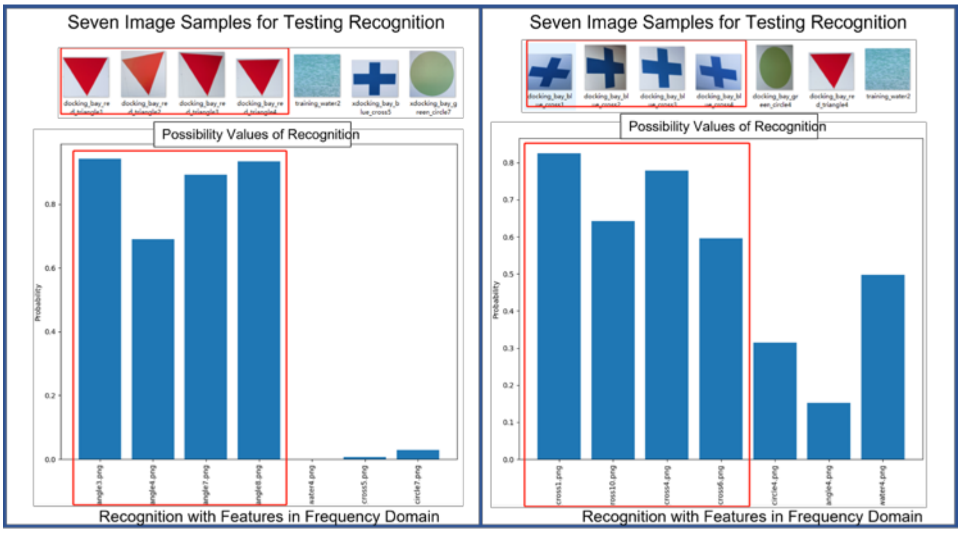


Figure 16. Two Examples of Results of Recognition Using Feature Vectors in Frequency Domain.

By now, people may ask whether the proposed new principle could work well with other more complex entities. For the sake of responding to such doubt, we give two more examples of results which make use of the feature vectors in frequency domain to do cognition and recognition. For each reference entity (e.g., car and dog), the cognition module

doi:10.20944/preprints202306.1313.v1

is trained with ten image samples while the recognition process is tested with six image samples as input.

In Figure 17, we show the experimental results of cognizing and recognizing cars in frequency domain. The results are judged to be very good.

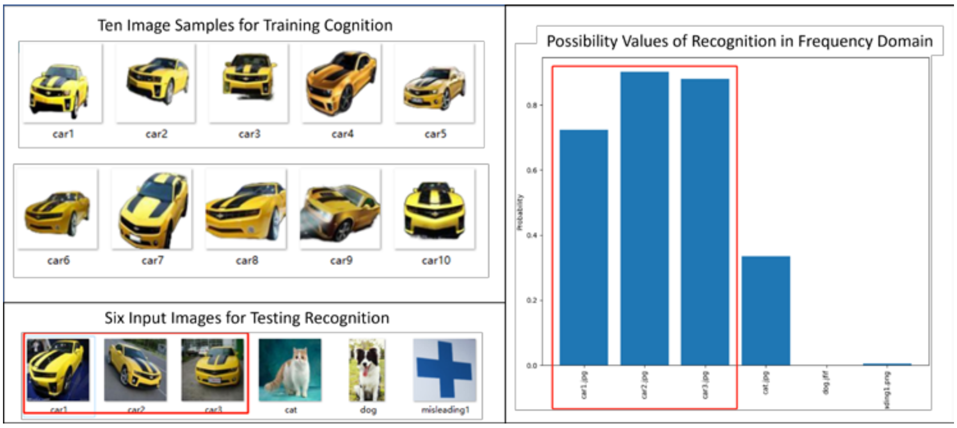


Figure 17. Example of Cognizing and Recognizing Cars in Frequency Domain.

In Figure 18, we show the experimental results of cognizing and recognizing dogs in frequency domain. The results are also judged to be very good.

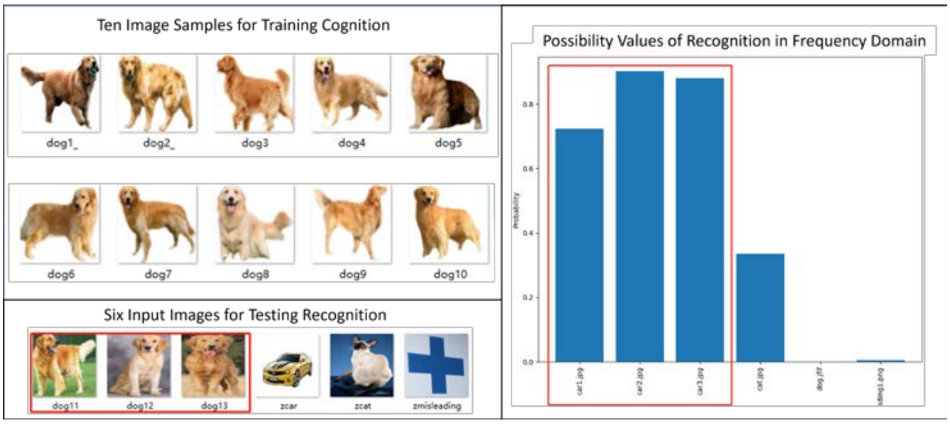


Figure 18. Example of Cognizing and Recognizing Dogs in Frequency Domain.

11.7. Results of Stereo Matching

Mathematically, a pair of images is good enough to validate a stereo matching algorithm. In practice, a pair of images could come from a binocular vision system which is normally named as stereovision system. Alternatively, a pair of images could come from a mobile monocular vision system. Since we use the image dataset from the public domain, it is easier for us to take two images from an image sequence captured by a mobile camera.

Here, we share one example of results in Figure 19, Figure 20, and Figure 21. In Figure 19, we let the stereovision system undergo the cognition process in which ten sample images of a floating post are used to train the RCE neural network inside the cognizer allocated to learn the red floating post. After the training of the cognizer's RCE neural network, the so-called stereovision system is ready to enter the recognition process which takes any set of new images as input.

In Figure 19, seven image samples are selected for testing the validity of trained RCE neural network. The possibility values show good outcome from the recognition process.

doi:10.20944/preprints202306.1313.v1

Now, we could start the process of stereo matching. As shown in Figure 20, the first step is to do image sampling. When the so-called left image is sampled into a matrix of 4x7 image samples, the occurrence of a red floating post could be recognized. Please take note that the image sample of this recognized occurrence is named as image sample 1a by our testing program.

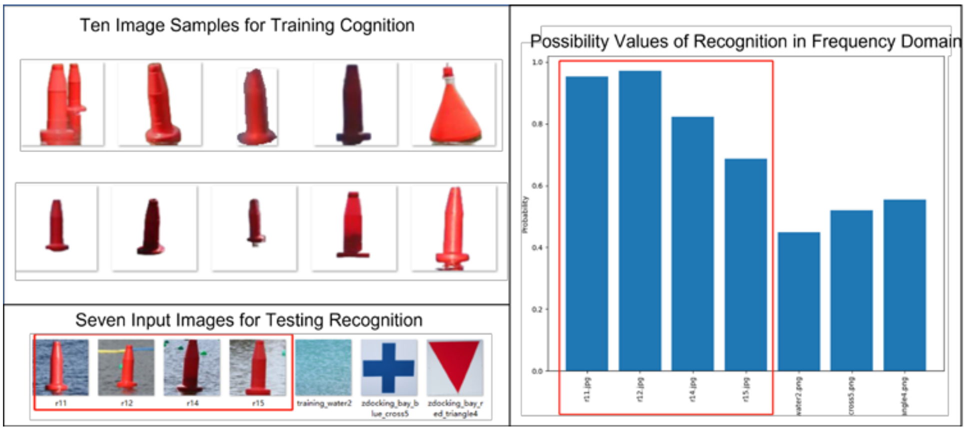


Figure 19. Results of Cognizing and Recognizing Red Floating Posts in Frequency Domain.

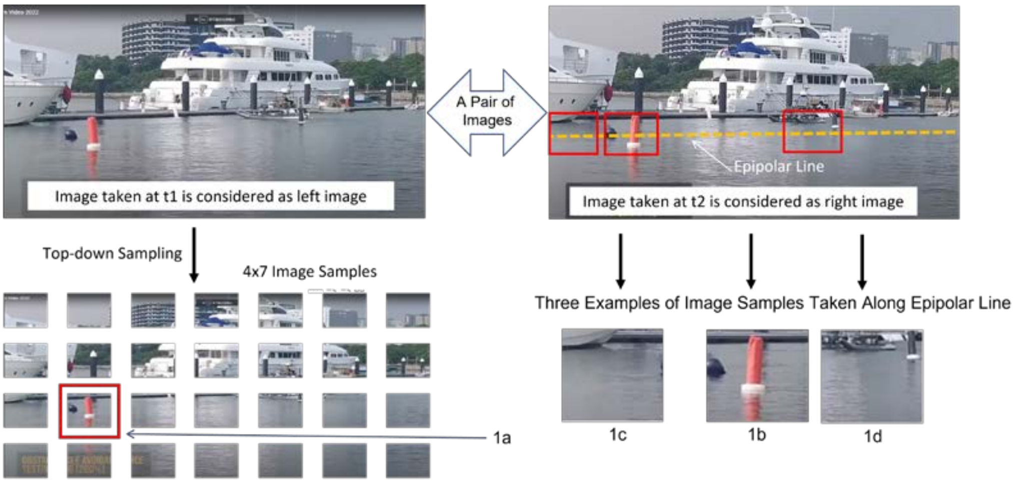


Image taken at t1 is considered as left image
A Pair of Images
Epipolar Line
Image taken at t2 is considered as right image
Top-down Sampling
4x7 Image Samples
1a
Three Examples of Image Samples Taken Along Epipolar Line
1c
1b
1d

Figure 20. Results of Testing Recognition with Seven Image Samples, after Doing Cognition with Ten Image Samples Which Have Certain Level of Intended Variations for the Purpose of Appreciating Robustness.

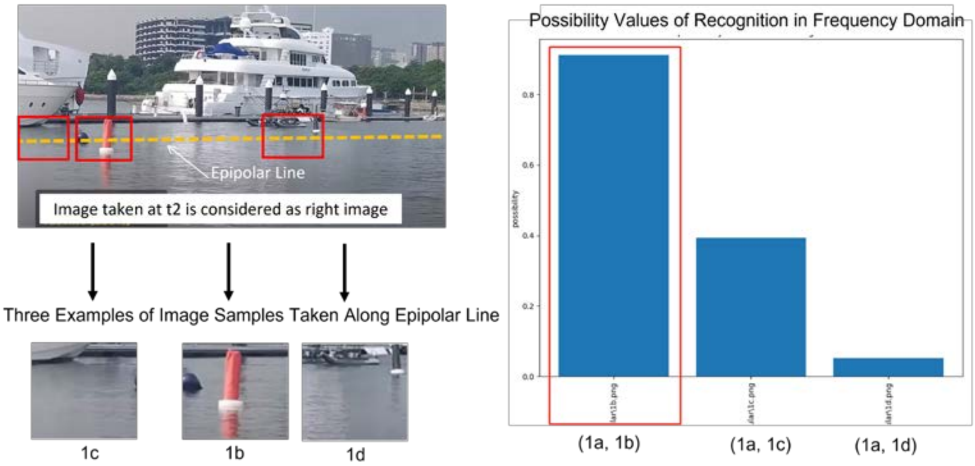


Figure 21. Results of Stere Matching Among Three Pairs: (1a, 1b), (1a, 1c), and (1a, 1d).

Subsequently, in the so-called right image, we could determine a line (i.e., equivalent to epipolar line) which will guide the search for the best match candidate.

For the purpose of illustration, we take three image samples to compute the stereo matching results. Among these three image samples, image sample 1b is the best match. In general, the best match is the one which maximizes the possibility value (i.e., computed by Equation 21 with $\sigma_0 = 10$) between image sample 6a in left image and all the possible image samples in right image. Figure 21 shows the possibility values computed for three pairs of possible matches, which are (1a, 1b), (1a, 1c), and (1a, 1d). Clearly, pair (1a, 1b) stands out to be the best match.

12. Conclusions

In this paper, we have described the details of the key steps in a proposed new principle which aims at achieving robust stereo matching in human-like stereovision. The main idea is to undertake stereo matching at a cognitive level. The significant contributions from this paper include: First, the introduction of a top-down sampling strategy will lighten the burden of subsequent processes in stereovision. This is because it will provide better versions of image samples, which will in return diminish the chance of committing errors by the subsequent processes in stereovision. Secondly, we advocate the process of feature extraction in both time domain and frequency domain. In this way, key characteristics of a visual entity will be able to be preserved as much as possible. Especially, we have highlighted the importance of Fourier series and Fourier coefficients in the process of extracting visual features from images. Thirdly, we have shown the important difference between artificial neural network and RCE neural network. Most importantly, we have introduced the possibility function to improve RCE neural network so as to make it a better way to support the process of cognition (including deep learning) as well as the process of recognition. Fourthly, we have introduced the inverse strategy of template matching. This is a better solution to cope with the problem of partial views due to occlusions or mis-aligned sampling of images. Last, but not the least, the key steps in the proposed new principle have been validated by experiments with real image data under the context of maritime RobotX challenge. The obtained results are very encouraging. We hope that more results and progress will emerge in this new direction of research.

Supplementary Materials: Nil.

Author Contributions: The first author is the contributor of the main ideas and the writer of the main text in this paper. The second and third authors are the contributors of the paper's implementations and experimentations which include data preparation and result presentation.

Funding: This research was funded by the Future Systems and Technology Directorate, Ministry of Defense, Singapore, grant number PA9022201473.

Institutional Review Board Statement: Not applicable".

Data Availability Statement: All data related to this paper will be made available upon request. The access to the data is subject to the data protection policy of Nanyang Technological University, Singapore.

Acknowledgments: We would like to acknowledge the financial support from the Future Systems and Technology Directorate, Ministry of Defense, Singapore, to NTU's RobotX Challenge team.

Conflicts of Interest: There is no conflict of interest.

References

1. Xie, M. Hu, Z. C. and Chen, H. *New Foundation of Artificial Intelligence*. World Scientific, **2021**.
2. Cassinelli, A. Reynolds, C. and Ishikawa, M. Augmenting Spatial Awareness with Haptic Radar, *IEEE International Symposium on Wearable Computers*, 2006, pp. 61-64.
3. Li, Y. and Ibanez-Guzman, J. Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems, *IEEE Signal Processing Magazine*, 2020, Vol. 37, No. 4, pp. 50-61.

4. Rashidi, A. Fathi, H. and Brilakis, I. Innovative Stereo Vision-Based Approach to Generate Dense Depth Map of Transportation Infrastructure, *Transportation Research Record*, 2011, Volume 2215, Issue 1.
5. Xie, M. Key Steps Toward Development of Humanoid Robots, *25th International Conference on Climbing and Walking Robots, Robotics in Natural Settings, Lecture Notes in Networks and Systems*, Springer, 2022.
6. Xie, M. and Velamala, S., Maritime Autonomous Vessels: A Review of RobotX Challenge's Works, *Journal of Technology and Social Sciences*, 2018, Vol.2, No.2, pp.7-14.
7. Gordon, I. E. *Theories of Visual Perception*, 3rd Edition, Psychology Press, 2004
8. Bekey, G. A. *Autonomous Robots: From Biological Inspiration to Implementation and Control*, The MIT Press, 2005.
9. Roberts, D. A. and Yaida, Sho. *The Principles of Deep Learning Theory*, The Cambridge University Press, 2022.
10. Wu, X. W. Sahoo, D. and Hoi, S. C. H. Recent advances in deep learning for object detection, *Neurocomputing*, 2020, Volume 396, Pages 39-64,
11. P. Rogister, P. Benosman, R. Ieng, S. H. Lichtsteiner, P. and Delbruck, T. Asynchronous Event-Based Binocular Stereo Matching, IEEE Transactions on Neural Networks and Learning Systems, 2012, Vol. 23, No. 2, pp. 347-353.
12. Yang, G. S. Manela, J. Happold, M. and Ramanan, D. Hierarchical Deep Stereo Matching on High-Resolution Images, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 2019, pp. 5515-5524.
13. Bleyer, M. and Breiteneder, C. *Stereo Matching: State-of-the-Art and Research Challenges*, in Edited Book of Advances in Computer Vision and Pattern Recognition, Springer, 2013, pp 143–179.
14. Chang, J. R. and Chen, Y. S. Pyramid Stereo Matching Network, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2018, pp. 5410-5418.
15. Xie, M. A Cooperative Strategy for The Matching of Multi-level Edge Primitives, *Image and Vision Computing*, 1995, Vol. 13, No. 2, pp. 89-99.
16. Medioni, G. and Nevatia, R. Segment-based stereo matching, *Computer Vision, Graphics, and Image Processing*, 1985, Volume 31, Issue 1, Pages 2-18,
17. Zhang, Y. N. and Gerbrands, J.J. Method for matching general stereo planar curves, *Image and Vision Computing*, 1995, Volume 13, Issue 8, Pages 645-655.
18. Wang, Z. F. and Zhi-Gang Zheng, Z. G. A region based stereo matching algorithm using cooperative optimization, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
19. L. Li, L. Fang, M. Yin, Y. Lian, J. and Wang, Z. A Traffic Scene Object Detection Method Combining Deep Learning and Stereo Vision Algorithm, *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2021, pp. 1134-1138.
20. Yin, X. M. Guo, D. and Xie, M. Hand image segmentation using color and RCE neural network, Robotics and Autonomous Systems, 2001, Volume 34, Issue 4, Pages 235-250.
21. Cooper, P. W. The hypersphere in pattern recognition", *Information and Control*, 1962, Vol. 5, pp. 324-346.
22. Morgan, D.P. and Scofield, C.L. ANN Keyword Recognition, *In Neural Networks and Speech Processing, The Springer International Series in Engineering and Computer Science*, 1991, Vol 130.
23. Cooper, L. N. *How We Remember: Toward an Understanding of Brain and Neural Systems*, World Scientific, 1995.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.